

# Claims

[c1] 1. A virtual Network Attached Storage (NAS) translator comprising:

a network interface that receives requests from clients, and receives server-data from a plurality of NAS appliance servers, each of the plurality of NAS appliance servers storing files and having a file system and a network connection;

wherein the server-data from the plurality of NAS appliance servers is carried over a network in network packets;

wherein the requests from clients are carried over the network in network packets;

wherein the requests from clients are messages using a networked-file-protocol, the requests including initial requests that contain a file name and a directory-path locator;

a hash engine, receiving the file name and a directory-path locator from the network interface, the hash engine generating a hashed-name key having a fixed-length;

a translation table with a plurality of entries that each contain a storage key, file meta-data, and a unique identifier, the entries not storing the file name;

wherein the meta-data includes a server identifier that indicates a server storing the file in the plurality of NAS appliance servers;

a comparator, receiving the hashed-name key and the storage key, for selecting a matching entry in the translation table, the matching entry having a storage key that matches the hashed-name key; and

a reply generator, receiving the unique identifier for the matching entry from the translation table, for generating a reply to the client that is sent through the network interface to the client, the reply containing a virtual file handle using the unique identifier.

- [c2] 2.The virtual NAS translator of claim 1 wherein the file name has a variable length and the hashed-name key and the storage key are each fixed-length values having a same fixed length.
- [c3] 3.The virtual NAS translator of claim 2 wherein the same fixed length is 6 bytes and the variable length is from 1 to 256 bytes.
- [c4] 4.The virtual NAS translator of claim 1 wherein the requests from clients also include secondary requests that contain the virtual file handle received by the client from the replay generator;  
further comprising:

a secondary translator, receiving the virtual file handle from the client, the secondary translator extracting a client unique identifier from the virtual file handle and searching the translation table for a matching entry that has the unique identifier matching the client unique identifier; and

a request forwarder, coupled to the network interface, for substituting a native file handle for the matching entry for the virtual file handle to generate a request to a NAS appliance server in the plurality of NAS appliance servers;

whereby virtual file handles from clients are replaced with native file handles by the virtual NAS translator before being forwarded to the plurality of NAS appliance servers.

[c5] 5. The virtual NAS translator of claim 4 wherein the translation table comprises:

a hashed-key translation table with entries selectable by matching the hashed-name key to the storage key in the matching entry;

wherein entries in the hashed-key translation table are also selectable by matching an input unique identifier to the unique identifier stored in the matching entry;

an open-file translation table having entries that store a unique identifier, meta-data, and a native file handle;

wherein the secondary translator finds a matching entry in the open-file translation table and outputs the native file handle from the matching entry.

- [c6] 6. The virtual NAS translator of claim 5 wherein the metadata for a selected entry in the hashed-key translation table further comprises:
  - server-handle pairs that comprise:
    - a network address of a selected NAS appliance server that stores a selected file referenced by the selected entry; and
    - a native file handle that a file system on the selected NAS appliance server uses to locate the selected file on the selected NAS appliance server;
  - wherein when the selected file is stored on a first server and a second server in the plurality of NAS appliance servers, the server-handle pairs include a first server-handle pair for a first copy of the selected file on the first server, and a second server-handle pair for a second copy of the selected file on the second server,
    - wherein server-handle pairs stored in the selected entry can locate redundant copies of files on different NAS appliance servers.
- [c7] 7. The virtual NAS translator of claim 5 further comprising:
  - a collision-resolution block storing colliding entries for a

first file and a second file that have a same storage key but have different file names or directory-path locators; wherein colliding entries comprise the same storage key, a first file name of the first file, a first unique identifier for the first file, a second file name of the second file, and a second unique identifier for the second file, wherein when the hash engine generates a hashed-name key that matches the same storage key, the collision-resolution block is accessed.

- [c8] 8.The virtual NAS translator of claim 5 wherein the hash engine executes a cryptographic hash function or a pseudo-cryptographic hash function.
- [c9] 9.The virtual NAS translator of claim 1 further comprising:
  - a secondary hash engine, receiving the hashed-name key from the hash engine, for generating a locator key; wherein the locator key selects a bucket of entries from the translation table;
  - wherein the comparator compares the hashed-name key to storage keys from the bucket of entries but does not compare to storage keys from un-selected buckets of entries,
  - whereby buckets of entries are selected by the locator key.

[c10] 10. The virtual NAS translator of claim 1 further comprising:

a counter that generates the unique identifier when a new entry is loaded into the translation table, wherein the virtual file handles are generated from the counter and not from the native file handles.

[c11] 11. The virtual NAS translator of claim 1 wherein each NAS appliance server comprises:

a server network connection;

a network–file layer that processes messages using the networked–file–protocol;

a file system having a native translation table that receives native file handles and generates physical file addresses;

a disk that stores files that are accessible by the physical file addresses.

[c12] 12. The virtual NAS translator of claim 11 wherein each NAS appliance server further comprises:

a Transport–Control–Protocol/Internet Protocol (TCP/IP) layer, coupled to the network connection and to the network–file layer, for extracting and encapsulating messages sent over the server network connection as network packets.

[c13] 13. The virtual NAS translator of claim 1 wherein the net–

work connection further comprises a Transport–Control–Protocol/Internet Protocol (TCP/IP) layer; wherein the network packets are TCP/IP packets.

- [c14] 14.The virtual NAS translator of claim 13 wherein the directory-path locator is a parent virtual file handle or a path name of a parent directory.
- [c15] 15.The virtual NAS translator of claim 14 wherein the networked–file–protocol is a Network–File–System (NFS) or a Common–Internet–File–System (CIFS).
- [c16] 16.A method for translating requests from a client for accessing files on a plurality of servers comprising:
  - receiving from a network a file-opening request from a client that contains a file name and a parent–directory virtual file handle;
  - combining the file name and the parent–directory virtual file handle into a combination name;
  - compressing the combination name with a cryptographic hash function to generate a hashed–name key;
  - searching a first table of file–translation entries for a matching entry that has a storage key that matches the hashed–name key;
  - reading a unique identifier from the matching entry;
  - generating a virtual file handle that includes the unique identifier;

sending the virtual file handle back to the client over the network in response to the file-opening request; receiving from the network a file-access request from the client that contains the virtual file handle; extracting a client-inputted unique identifier from the virtual file handle from the client; searching the first table for the matching entry that has the unique identifier that matches the client-inputted unique identifier; reading a server identifier from the matching entry; obtaining a native file handle from a server using the server identifier and the file-access request from the client; replacing the virtual file handle in the file-access request with the native file handle to generate a translated file-access request; and sending the translated file-access request over the network to a selected server in the plurality of servers; the selected server using the native file handle to access a selected file stored on the selected server, the selected server sending data from the selected file back to the client over the network, whereby file-access requests are translated from virtual file handles to native file handles.

[c17] 17. The method of claim 16 wherein the network carries

Transport–Control–Protocol/Internet Protocol (TCP/IP) packets; wherein the file-opening request, the file-access request, and the translated file-access request are messages for a network–file–system (NFS) or a Common–Internet–File–System (CIFS) protocol layer on the selected server and on the client.

- [c18] 18. A computer–program product comprising:
  - a computer–usable medium having computer–readable program code means embodied therein for translating network–file–access messages, the computer–readable program code means in the computer–program product comprising:
    - network connection means for transmitting and receiving packets from a remote client and from a remote server in a plurality of Network Attached Storage (NAS) appliance servers each having a file system;
    - concatenation means, receiving a file name and a directory handle from the remote client, for generating a combined name–handle;
    - hash means, receiving the combined name–handle, for generating a hashed–name key by executing a cryptographic hash function;
  - wherein the hashed–name key has a fixed length and the combined name–handle has a variable length;

table means for storing a plurality of file-translation entries, each file-translation entry having a storage key, a unique identifier, and a server identifier used to obtain a native file handle for a file stored on the remote server; first search means, responsive to the hashed-name key and coupled to the table means, for finding a matching entry in the table means, the matching entry having a storage key that matches the hashed-name key; initial reply means for generating a reply to the remote client that contains a virtual file handle for the file, the virtual file handle being generated from the unique identifier from the matching entry; second search means, responsive to a second request from the remote client, the second request containing the virtual file handle generated from the initial reply means and sent to the remote client, for searching the table means for the matching entry that has the unique identifier matching an extracted unique identifier extracted from the virtual file handle from the remote client; and second reply means for generating a forwarded request to the remote server, the forwarded request containing the native file handle from the matching entry.

[c19] 19. The computer-program product of claim 18 wherein the table means comprises:

hashed-key translation table means for storing entries with the storage key, the unique identifier, and the server identifier but not the file name and not the native file handle;

open-file translation table means for storing entries with the unique identifier and the native file handle but not the storage key and not the file name;

collision-resolution block means for storing entries containing a first file name of a first file and a second file name of a second file that generate a same hashed-name key, a first unique identifier for the first file and a first native file handle for the first file, and a second unique identifier for the second file and a second native file handle for the second file.

[c20] 20. The computer-program product of claim 19 further comprising:  
counter means for generating the unique identifier when a new entry is loaded into the table means.